A

Batch Mode Active Learning for Networked Data

Lixin Shi, Tsinghua University
Yuhang Zhao, Tsinghua University
Jie Tang*, Tsinghua University

We study a novel problem of batch mode active learning for networked data. In this problem, data instances are connected with links and their labels are correlated with each other, and the goal of batch mode active learning is to exploit the link-based dependencies and node-specific content information to actively select a batch of instances to query the user for learning an accurate model to label unknown instances in the network. We present three criteria (i.e., minimum redundancy, maximum uncertainty and maximum impact) to quantify the informativeness of a set of instances, and formalize the batch mode active learning problem as selecting a set of instances by maximizing an objective function which combines both link and content information. As solving the objective function is NP-hard, we present an efficient algorithm to optimize the objective function with a bounded approximation rate. To scale to real large networks, we develop a parallel implementation of the algorithm. Experimental results on both synthetic data sets and real-world data sets demonstrate the effectiveness and efficiency of our approach.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Text Mining; I.2.6 [**Artificial Intelligence**]: Learning

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Batch mode active learning; Network classification; Combine link and content

ACM Reference Format:
Shi, L., Zhao, Y., Tang, J. 2011. Batch Mode Active Learning for Networked Data. ACM Trans. Embedd. Comput. Syst. V, N, Article A (January YYYY), 26 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Machine learning algorithms often suffer from insufficiently labeled training data. Active learning aims to, not only, as usual, construct an accurate classifier, but also minimize the number of labeled instances by actively selecting a few number of instances to query the user. Traditionally, this problem is addressed in a single mode, i.e., the active learning algorithm queries the user k times, and each time queries one instance for its label. Following this thread, considerable research has been conducted on how to select the best example to query in each time [Rajan et al. 2010; Beygelzimer et al. 2009; Harpale and Yang 2008].

Author's addresses: Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. Lixin Shi (shilixinhere@gmail.com), Yuhang Zhao (zhaoyh630@gmail.com), Jie Tang (jietang@tsinghua.edu.cn)
*corresponding author: Jie Tang (jietang@tsinghua.edu.cn)
The work is supported by the Natural Science Foundation of China (No. 61073073), Chinese National Key Foundation Research (No. 60933013, No.61035004), and 973 Research Project (No. 2007CB310803).

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

Recently, there has seen a new direction of machine learning field, that is how to learn an accurate model to classify the networked data, e.g., the linked Web pages and the friendship network. A number of models have been proposed, such as Conditional Random Fields [Lafferty 2001], Continuous Bayesian network [Nodelman et al. 2003], Factor Graph models [Kschischang et al. 2001; Tan et al. 2010], Collective Learning [Jensen et al. 2004], and Semi-supervised Learning over graphs [Zhu 2005]. A few models also try to combine the link and the node-specific content information in a unified framework for active learning [Macskassy 2009; Zhu et al. 2003; Rattigan et al. 2007; Bilgic and Getoor 2010]. However, two important issues have been largely ignored in existing work. First, almost all algorithms for learning/classifying the networked data are computationally intensive, due to iterative selection and re-training the model. Suppose a machine needs to query the user k times, when the user inputs a label for the queried instance, she/he may have to wait for a long time for the next query, which is obviously undesirable. Second, most methods only consider the single mode. It is inefficient to re-train the model after querying only one instance and also there might exist information redundancy between the selected instances in different iterations. In this work, we aim to answer the question: how to actively select a set of instances from the networked data and query the users in a batch mode?

Motivating Example We refer to this problem as the *Batch Mode active learning (BMAL)* problem for networked data. A simple baseline method to address this problem is to design a metric to measure the informativeness of each candidate instance and then select instances with the highest informative scores. However, this method cannot guarantee an optimal solution, because: (1) simply accumulating candidate instances with the highest informativeness score does not necessarily mean the instance set also has the highest informativeness score; (2) such a method cannot take advantage of the link information.

In this paper, we try to conduct a systematic investigation of the problem of batch mode active learning for the networked data. Figure 1 demonstrates our problem. Suppose we are given a data set with only one labeled sample (say, with positive label), and the samples are connected in a two-dimension space. We try to select two samples to query the user in a batch mode. If we only consider the content information, then we would tend to select the two center nodes (top-right figure); while if we only consider the link information, we would select two samples with the most links (middle-left figure). By combining the link and content information, our BMAL method suggests two different samples (middle-right figure). The bottom two figures show how a machine learning algorithm updates the classification model when the user provides labels to the queried samples (suppose one positive and one negative).

Challenges and Contributions Thus the problem becomes how to quantify the informativeness of each candidate instance by combining the content and link information, and how to actively select a number of instances with the highest informativeness and as well the minimum redundancy among them. Comparing with existing work, there are several unique challenges for the batch mode active learning problem.

— First, as the problem of finding the most informative instances is NP-hard, how to formulate the problem in a unified framework is a challenging problem.
— Second, a central problem is how to design appropriate criteria to quantitatively measure the informativeness of the data.
— Third, the active learning algorithm should be efficient, in particular considering the increasing scale of the networked data on the Web.

To this end, we formally define the problem and propose a general batch mode active learning framework. Specifically, we propose three criteria to respectively capture the

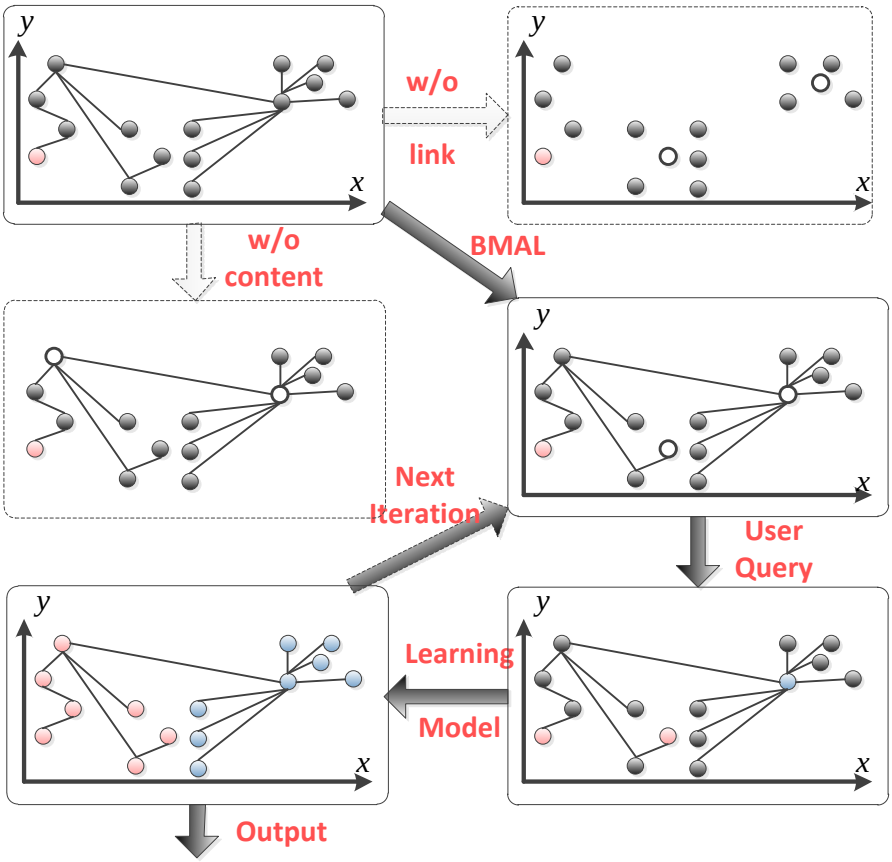


Fig. 1. An example illustration for the problem of batch mode active learning (BMAL) for networked data. The top-left figure plots the input networked data, where pink circles indicate instances with positive labels, blue ones indicate instances with negative labels, and gray ones indicate unknown labels. "w/o link" stands for active learning not considering links; "w/o content" stands for active learning without the content information; and "BMAL" stands for active learning by the proposed method, which considers both the content and link information.

maximum uncertainty, maximum impact, and minimum redundancy (which will be explained in section 2.2). Existing batch mode active learning frameworks are shown to satisfy only one or two of them. We present an objective function based on the criteria and prove that our method respects all three of them both intuitively and theoretically. An efficient algorithm is designed to solve the objective function and theoretical analysis for the approximation rate of the algorithm is given. We conduct experiments on both synthetic data sets and real-world data sets to validate the effectiveness and efficiency of our approach. Experimental results show that the proposed approach clearly outperforms (up to 6%) several baseline methods of single mode active learning and batch mode active learning for the networked data.

Organization The rest of this paper is organized as follows: Section 2 defines the batch mode active learning problem, determines the three criteria, and introduces a basic framework of random walk. Section 3 discusses our model in details, section 4 presents the active selection algorithm as well as parallelization to it, and section 5

presents the experiment results. Finally section 6 discusses some related work and section 7 concludes our work.

2. PRELIMINARIES

2.1. Problem Definition

The input of the BMAL (batch mode active learning) problem is defined as an (un)directed network (graph) $\mathcal{G} = (V, E)$, where vertices correspond to the data instances and the edge implies relationship between data instances, e.g., friendship or the citation relationship. The set of data instances V is comprised of n unlabeled data instances $\mathcal{U} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ with index set $U = \{1, 2, \cdots, n\}$ and l labeled data instances $\mathcal{L} = \{(\boldsymbol{x}_{n+1}, y_{n+1}), \cdots, (\boldsymbol{x}_{n+l}, y_{n+l})\}$ with index set $L = \{n+1, \cdots, n+l\}$, where $l \ll n$ under most circumstances, $\boldsymbol{x}_i (1 \leq i \leq n+l)$ is the observed (feature) vector, and $y_i \in \{0, 1\}$ is the classification label of the i-th data instance.

Following this, we can define the problem of batch model active learning for the networked data as follows:

Problem 1. **BMAL(Batch Mode Active Learning):** Given such a system $(\mathcal{U}, \mathcal{L}, \mathcal{G})$ and an integer k, how to select a set of k ($k \ll n$) instances $S \subset \mathcal{U}$ to query their classification labels, so that we can maximally improve the quality of the learned classification model for the networked data based on the queried data instances?

To solve this problem, a general objective function is defined for active selection of the data instances from the network:

$$\begin{aligned} S &= \operatorname*{argmax}_{S \subseteq \mathcal{U}, |S| \leq k} \{Q(S)\} \\ \text{with} \quad & Q : 2^{\mathcal{U}} \to \mathbb{R}, \text{ in system } (\mathcal{U}, \mathcal{L}, G) \end{aligned}$$

Thus, the BMAL problem is equivalent to the set function optimization problem. Now, the task is to well define the function $Q(S)$ and to design an efficient algorithm to maximize $Q(S)$. Please note that if there is no link information in the network, the problem degrades to a traditional data classification problem.

2.2. Three Criteria

We first define three criteria to measure the informativeness of selected candidate instances, as a guide of designing $Q(S)$:

— *Maximum Uncertainty*: We are always interested in choosing instances of uncertainty to query the user. One intuitive method for the binary classification (positive vs. negative) is to choose instances with posterior probabilities of being positive close to 0.5.
— *Maximum Impact*: Selected instances should have the maximum impact on other unknown instances. We do not expect to select instances which are isolated in the instance space, e.g., outliers. The impact can be considered from two aspects: the content similarity and the structure similarity among instances.
— *Minimum Redundancy*: The selected instances should be diversely distributed in the instance space. In other words, we need to minimize the information overlap between the selected instances.

Recently, a few methods have been proposed, which also try to actively find a batch of instances. To show that existing batch mode active learning methods are not suitable

in our BMAL framework, we first evaluate them under these criteria before proposing our method.

The first kind of batch mode active learning method is based on SVM. [Tong and Chang 2001] suggests to choose instances that are close to the decision boundary of SVM classifier. However the method may result in undesirable redundancy in the selected instances. This problem is pointed out in [Brinker 2003] and a diversity is added in [Hoi et al. 2008; Brinker 2003]. Another kind of batch mode active learning methods employs the Fisher information matrix to measure informativeness [Hoi et al. 2006; Steven et al. 2009]. It maximizes uncertainty and implicitly minimizes redundancy, as stated in [Hoi et al. 2006]. However, both of them ignore the impact of the selected instances on other unlabeled instances in the network, thus the selected instances may have no relationship (links) with the other ones. As a result, the user's labeling efforts on these instances cannot be optimally leveraged to infer labels of the other unlabeled instances. There are also some other batch mode active methods, such as [Joshi et al. 2010; Xu et al. 2009]. Different from existing work, we define an objective function strictly based on these three criteria, which is demonstrated to be essential by both theoretical analysis and empirical evaluation.

2.3. A Framework of Random Walk

We introduce a framework of random walk [Zhu et al. 2003], which is designed for semi-supervised learning, i.e., learning a classification model with a few labeled data L and a large number of unlabeled data U. Specifically, in this framework, a $(n+l) \times (n+l)$ pair-wise similarity matrix W is introduced, where each element w_{ij} measures the similarity between feature vector $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The similarity can be defined in different ways, for example, the radial basis function (RBF):

$$w_{ij} = \exp\left(-\frac{1}{\sigma^2}||\boldsymbol{x}_i - \boldsymbol{x}_j||^2\right) \tag{1}$$

Suppose the expectation of $y_i (i \in U)$ is f_i, i.e., the probability that $y_i = 1$ under Bernoulli distribution. There is a nice interpretation of f_i based on the random walk theory. We first normalize the weight matrix W to be $\tilde{W}$, where

$$\tilde{w}_{ij} = \frac{w_{ij}}{\sum_{k \neq i} w_{ik}}, i \neq j$$

and $\tilde{w}_{ij} = 0, i = j$. A random walk system is then defined over all the data instances (points).

— The transition probability from an unlabeled data point $i (i \in U)$ to any data point $j (j \in U \cup L)$ is $\tilde{w}_{ij}$;
— All the labeled data points are the absorbing nodes: it transforms to itself with probability 1.

It can be seen from the definition above that all transformations in the system will eventually go into a self-loop at a labeled point. f_i is the probability that a particle, starting from unlabeled data point i, will eventually get looped in a labeled point with label 1. A theoretical verification is given in [Zhu et al. 2003]. Here we employ this framework to calculate f_i: the transition matrix defined by the random walk system is therefore

$$P = \begin{bmatrix} A & B \\ O & I \end{bmatrix}$$

Table I. Variables Used in Our Model

Symbol	Description
$\mathcal{U}$	The set of unlabeled data instances
$\mathcal{L}$	The set of labeled data instances
$\mathcal{G}$	The network representing the relationships between instances
$\boldsymbol{x}_i$	The feature vector of instance i
y_i	The label of instance $i \in \mathcal{L}$
W	The similarity matrix
P	The transition matrix in the random walk framework
$\boldsymbol{f}_u$	Expectation vector of instances in unlabeled data set
$\boldsymbol{f}_l$	Expectation vector of instances in labeled data set
f_i	The expectation of instance i's label
S	The set of instances to be selected
k	The number of instances to be selected

where O is the zero matrix, I is the identity matrix. And the matrix $[\ A\ \ B\ ]$ is the first n lines of $\tilde{W}$, which correspond to the unlabeled data instances. Specifically, A is the transition matrix between points in unlabeled data sets, and B is the transition matrix from unlabeled data instances to labeled ones.

Correspondingly, we define $\boldsymbol{f}_u$ and $\boldsymbol{f}_l$ as the expectation vector of all unlabeled data points and labeled data points, i.e.,

$$\boldsymbol{f}_u = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \boldsymbol{f}_l = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ \vdots \\ y_{n+l} \end{bmatrix}$$

Then we have

$$\begin{bmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_l \end{bmatrix} = \begin{bmatrix} A & B \\ O & I \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_l \end{bmatrix} \tag{2}$$

The solution can be given by

$$\boldsymbol{f}_u = (I - A)^{-1} B \boldsymbol{f}_l$$

This framework will work as a underlying model of the BMAL method. As a summary, table I lists the notations used throughout this paper.

3. THE PROPOSED APPROACH

3.1. Objective Function

Our basic idea is to define an objective function $Q(S)$ to combine the three criteria (Cf. Section 2.2) together. To this end, we define two functions, $H(S)$ and $C(S)$, respectively represent the maximum uncertainty and the maximum impact. We will show later that this combination will naturally satisfies the minimum redundancy criterion. The objective function is defined as a linear combination of the two functions, i.e.,

$$Q(S) = \alpha C(S) + (1 - \alpha) H(S), \quad 0 \leq \alpha \leq 1 \tag{3}$$

where α is a parameter to balance the importance of two functions.

Maximum Uncertainty We use entropy to measure the uncertainty of selected instances. Joint entropy would be difficult to compute, thus we use the summation of entropies over single data instances. The maximum uncertainty part is defined as the

$H(S)$ function in $Q(S)$:

$$H(S) = \sum_{i \in S} H(i) = \sum_{i \in S} f_i \log \frac{1}{f_i} + (1 - f_i) \log \frac{1}{1 - f_i} \tag{4}$$

where $H(i)$ is the entropy of data instance i and f_i is the expectation of instance i's label (Cf. Table I).

Maximum Impact The criterion is to measure how a selected instance can influence the other unknown instances. More accurately, it estimates to which extent labeling one instance can help classify the other unknown instances. This measurement of maximum impact comes from the classical nearest neighborhood classifier. The classifier classifies data instance $\boldsymbol{x}_i$ into the same class with labeled data instance $\boldsymbol{x}_j$ which has the highest impact (distance) on $\boldsymbol{x}_i$:

$$Class(\boldsymbol{x}_i) = y_j, \quad j = \operatorname*{argmax}_{j \in L} w_{ij}$$

From the view of Nearest Neighbor classifier, the classification result is more guaranteed if the impact is higher. That gives a direct motivation on the maximum impact measurement: to maximize the impact on a single unlabeled data instance $\boldsymbol{x}_i$, we can choose the data instances with the maximum impact over $\boldsymbol{x}_i$ from the candidates. So we can have a weighted function of summations over all these maximum values to measure the impact:

$$C(S) = \sum_{i \in U} s_i \max_{j \in L \cup S} w_{ij} \tag{5}$$

where s_i serves as a weight factor when counting the impact over instances in the unlabeled data set; w_{ij} indicates the similarity between instance i and j. There may be better choices other than choosing $s_i = 1$ for all unlabeled data instances, e.g. using entropy as the weight. Specifically,

$$s_i = H(i) = f_i \log \frac{1}{f_i} + (1 - f_i) \log \frac{1}{1 - f_i} \tag{6}$$

The point is that the use of entropy information here does not overlap with the entropy in $H(S)$. This is because different examples are checked by $C(S)$ and $H(S)$ in terms of entropy. To achieve a higher flexibility, we introduce a parameter β to further balance the importance of the two terms:

$$C(S) = \sum_{i \in U} (H(i))^{\beta} \left(\max_{j \in L \cup S} w_{ij} \right)^{1-\beta} \tag{7}$$

Minimum Redundancy In equation 3, we do not have a term to explicitly demonstrate the redundancy over the selected set. In this section, we will prove that the minimum redundancy criterion has already been implicitly satisfied in the definition of $Q(S)$. To start with, we give an intuitive impression about why maximizing $Q(S)$ will also minimize the redundancy. Given a data instance $i \in U - S$, let us define the dominant instance $dp(i)$ as

$$dp(i) = \max_{j \in S \cup L} w_{ij} \tag{8}$$

We will show that maximizing $Q(S)$ will cause the dominant instances to get diversely distributed. If two dominant instances in S are very close to each other, it is

likely that they have a similar impact on the other unknown instances, thus removing one of them from S will not cause $C(S)$ to decrease a lot. In other words, if we already have one of them, say, vertex i, in the selected set S, we would avoid choosing another similar vertex j in the future, because the increase on $Q(S)$ is limited.

Redundancy of the selected set can be measured by the following function

$$R(S) = \sum_{i,j \in S \cup L, i \neq j} w_{ij} \tag{9}$$

Theorem 3.1 explores the relationship between maximizing $C(S)$ and minimizing $R(S)$.

THEOREM 3.1. *If w is under the definition of RBF function, and S satisfies that $\forall j \in S, \exists i \in U - S, j = dp(i)$ we have the following estimation of $R(S)$:*

(1) $$R(S) \geq \frac{k+l}{n-k} \sum_{i \in U-S} \left(\frac{1}{w_{i,dp(i)}} \sum_{j \in S \cup L} w_{ij} \right) - (k+l)$$

(2) $$R(S) \leq \frac{k+l}{n-k} \sum_{i \in U-S} \left(\frac{1}{w_{i,dp(i)}} \sum_{j \in S \cup L} \sqrt{w_{ij}} \right) - (k+l)$$

Note that the requirement of S in the theorem is satisfied if we use the greedy algorithm suggested in section 4. The proof of theorem 3.1 is given in the appendix. From theorem 3.1, we can see that $R(S)$ is tightly bounded by terms $\frac{1}{w_{i,dp(i)}} \sum_{j \in S \cup L} w_{ij}$ and $\frac{1}{w_{i,dp(i)}} \sum_{j \in S \cup L} \sqrt{w_{ij}}$. Since maximizing $C(S)$ is equivalent to maximizing $\sum_{i \in U-S} w_{i,dp(i)}$, it approximately minimizes these two terms (because $\sum_{i \in U-S} \sum_{j \in S \cup L} w_{ij}$ and $\sum_{i \in U-S} \sum_{j \in S \cup L} \sqrt{w_{ij}}$ are close to constant when n is far larger than k).

In summary, the definition of $Q(S)$ in equation 3 serves as a measurement of maximum entropy and maximum impact, and it implies minimum redundancy both intuitively and theoretically. Further in Section 5.1, we will use a synthetic data set to empirically validate the necessity of these three criteria.

3.2. Combining Link Information

One of the challenges in BMAL is how to combine the link information into our approach. In this section, we introduce how to address this problem by integrating link information into a *similarity matrix*.

Similarity Matrix W Similarities between data instances form a $(n+l) \times (n+l)$ similarity matrix W. In our problem, the similarity measurement should satisfy the following properties (which are common and easy to satisfy):

— Larger value of w_{ij} indicates higher similarity.
— $0 \leq w_{ij} \leq 1$, moreover $w_{ij} = 1$ if and only if $\boldsymbol{x}_i = \boldsymbol{x}_j$.
— Similarity should be approximately transitive. That is, $\exists \lambda_1 \geq 1, \lambda_2 \leq 1$, $(w_{ij} w_{jk})^{\lambda_1} \leq w_{ik} \leq (w_{ij} w_{jk})^{\lambda_2}$. We call such similarity matrix (λ_1, λ_2)-transitive. This property is critical to ensure the diversity of selected instances (Cf. Appendix).

Eq. 1 is a simple similarity definition using RBF. There is one problem left in the definition, i.e., how to get the appropriate value of σ. Higher σ value will make the matrix ill-conditioned or even singular, while lower σ value will hurt the discriminative capacity of the similarity matrix W. A possible way to learn the parameter is to find σ

that minimizes average label entropy, i.e.,

$$\frac{1}{n}\sum_{i\in U} H(i)$$

Note the definition of $H(i)$ by Eq. 4 uses probability f_i, which is calculated from W. Interested readers please refer to [Zhu et al. 2003].

Combining Link Information Now, we introduce how to combine the link information into the similarity matrix. This can be done by extending the similarity measure with a link-based one such as pagerank.

Pagerank is an algorithm to estimate the importance of each node in a graph. In essence, it calculates a transition matrix based on the graph structure. The transition matrix can be used as the asymmetric similarity between nodes. Generally speaking, the pagerank model can be explained by a surfer randomly jumping in the graph:

— The surfer may jump to another node by following links with a equal or weighted probability.
— The surfer may randomly jump to any node with a probability proportional to similarity in feature space.

Suppose there is a well defined similarity matrix W that measures the impact solely in feature vector space. Now we want to integrate the link information into it, we can obtain a new definition $\widehat{W}$:

$$\hat{w}_{ij} = \epsilon\frac{1}{d_i}I(i,j) + (1-\epsilon)\frac{w_{ij}}{\sum_k w_{ik}}$$

where $0 \le \epsilon \le 1$, $I(i,j)$ is an indicator function standing for whether there is a link between instances i and j:

$$I(i,j) = \begin{cases} 1 & (i,j) \in E \\ 0 & (i,j) \notin E \end{cases}$$

and d_i is the degree of i, $d_i = \sum_{(i,j)\in E} 1$.

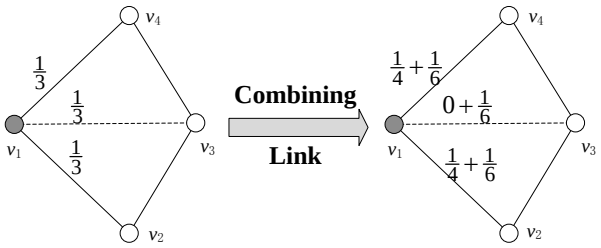


Fig. 2. An example of combining link information. Network graph $\mathcal{G}$ are shown in Solid lines.

In this way, the weight (transition probability) consists of two parts: the link information and the similarity. ϵ is a factor balancing the weight of these two parts. We take data instance (point) v_1 in figure 2 as an example. Originally the similarities between v_1 and the other 3 points are identical, i.e., $w_{12} = w_{13} = w_{14} = \frac{1}{3}$. Suppose $\epsilon = \frac{1}{2}$, since v_1 are connected to v_2 and v_4, the link parts of w_{12} and w_{14} are both $\frac{1}{4}$, while the link part of the w_{13} are zero. The final weights after combining link information are $w_{12} = w_{14} = \frac{5}{12}, w_{13} = \frac{1}{6}$, the $\frac{1}{4}$ difference are caused by links.

ALGORITHM 1: Greedy algorithm for batch mode active selection. The algorithm iteratively selects a sample that can maximize $Q(S)$

Input: U, L, G, $\mathbf{x}$, $\mathbf{y}$, k
Output: S, s.t.$|S| = k$
Calculate transition matrix P;
Calculate probability vector $\mathbf{p}$;
for $v \in U$ **do**
  $H[v] \leftarrow p_v \log \frac{1}{p_v} + (1 - p_v) \log \frac{1}{1-p_v}$;
  initialize: $C[v] \leftarrow 0, \max[v] \leftarrow 0$;
end
initialize: $S \leftarrow \emptyset$;
while $|S| < k$ **do**
  for $v \in U - S$ **do**
    $C[v] \leftarrow$ The summation over $j \in U$;
      of $(H[j])^{\beta} \left(\max\{\max[j], w(v,j)\}^{1-\beta}\right)$;
  end
  Find $v \in U - S$ to maximize:
    $\alpha C[v] + (1 - \alpha) H[v]$;
  update: $S \leftarrow S \cup \{v\}$;
  update: $\max[j] \leftarrow \max\{\max[j], w(v,j)\}$;
end

4. LEARNING ALGORITHM

Solving the objective function (Eq. 3) is NP-hard. Several greedy algorithms can be considered to approximate the optimal solution. However, most of them cannot theoretically guarantee an error bound. In this section, we present an efficient learning algorithm. More importantly, since the $Q(S)$ is designed as a monotonic submodular function, the solution of the learning algorithm can have a good error bound. We will give a theoretical analysis to the submodularity and monotonicity of the function $Q(S)$. Finally, to scale up to real large data set, we have developed a parallel implementation of the algorithm.

Algorithm 1 outlines the learning algorithm. The algorithm can be roughly divided into two parts: the first part are mainly matrix operations, calculating transition matrix P, probability vectors and entropy $H[v]$ for each instance v; the second part runs iteratively on the unlabeled data set $U - S$, with each time selecting a data instance v of the maximum score:

$$Q(S \cup \{v\}) = \alpha C[v] + (1 - \alpha) H[v] \tag{10}$$

Though we select all the samples one by one, this is a batch mode active selection method. Each time we greedily select a batch of samples to query users; while in the single mode active learning framework, users are queried each time when a single sample is selected.

Given the function $Q(S)$ being submodular and monotonic, the algorithm is guaranteed with an approximation rate of $(1 - \frac{1}{e})$, as shown in theorem 4.1.

THEOREM 4.1. *The approximation rate of the algorithm above is* $(1 - \frac{1}{e})$. *Specifically, let* S *be the output of the algorithm and* S^* *be the optimal solution, we have* $\frac{Q(S)}{Q(S^*)} \geq (1 - \frac{1}{e})$.

Proof of this theorem (the approximation rate) has been extensively studied, and can be found in [Nemhauser et al. 1978]. Here we give the proof of the monotonically submodularity property.

4.1. Proof of Submodularity and Monotonicity in $Q(S)$

Submodularity is an elegant property for set function optimization problems [Kawahara et al. 2009; Nemhauser et al. 1978]. Our defined function $Q(S)$ satisfies the submodular property.

THEOREM 4.2. *$Q(S)$ is submodular. That is, $\forall S_1 \subseteq S_2 \subseteq S$, $\forall v \notin S_2$,*

$$Q(S_1 \cup \{v\}) - Q(S_1) \geq Q(S_2 \cup \{v\}) - Q(S_2)$$

PROOF. Since $H(S) = \sum_{v \in S} H(v)$ is an additive function, which is submodular, we only have to prove that $C(S)$ is submodular.

Suppose $S_1 \subseteq S_2 \subseteq S$, then $C(S_1 \cup \{v\}) - C(S_1)$ is:

$$\sum_{i \in U} s_i \left(\max_{j \in S_1 \cup L \cup \{v\}} w_{ij} - \max_{j \in S_1 \cup L} w_{ij} \right)$$

Define $\Delta(i, S_1) = \max_{j \in S_1 \cup L \cup \{v\}} w_{ij} - \max_{j \in S_1 \cup L} w_{ij}$, we have $\Delta(i, S_1) \geq \Delta(i, S_2)$ for $\forall i \in U$:

— If $\Delta(i, S_1) = 0$, then there $\exists j \in S_1 \cup L, w_{ij} \geq w_{iv}$. Since $S_1 \subseteq S_2$, $\max_{j \in S_2 \cup L \cup \{v\}} w_{ij} = \max_{j \in S_2 \cup L} w_{ij}$ as well.

— Otherwise, w_{iv} is larger than w_{ij}, $\forall j \in S_1$. If $\Delta(i, S_2) = 0$, the proof is done; otherwise

$$\Delta(i, S_1) = w_{iv} - \max_{j \in S_1 \cup L} w_{ij} \geq w_{iv} - \max_{j \in S_2 \cup L} w_{ij} = \Delta(i, S_2)$$

So we have that $\sum_{i \in U} \Delta(i, S_1) \geq \sum_{i \in U} \Delta(i, S_2)$, that is, $C(S)$ is submodular. Therefore, we can obtain that $Q(S)$ is submodular. The defined function $Q(S)$ is also monotonic. □

THEOREM 4.3. *$Q(S)$ is monotonic.*

PROOF. $\forall S_1 \subseteq S_2 \subseteq S$, it's not hard to see that $H(S)$ is monotonic, and for $C(S)$,

$$C(S_1) = \sum_{i \in U} s_i \max_{j \in L \cup S_1} w_{ij} \leq \sum_{i \in U} s_i \max_{j \in L \cup S_2} w_{ij} = C(S_2)$$

Therefore, $Q(S)$ is monotonic. □

4.2. Parallelism

With the volume of the data set process increasing, algorithm 1 would suffer from the limitation of memory space and computation power with only one machine. Thus it is necessary to design a parallel algorithm for scaling up to real large data set. We have several strategies to improve the scalability of the algorithm. First, sparse representation and distributed storage are used to store the similarity matrix W. Second, we have implemented a parallel version of Algorithm 1. Specifically, we employ the MPI (Message Passing Interface) as the parallel programming model, which is a widely-used language-independent parallel library [Gropp et al. 1994].

The first time-consuming part of algorithm 1 is the matrix multiplication and inversion when computing probability matrix P. How to parallelize matrix operations has been a classical problem. We use Cannon Matrix-Matrix Multiplication Algorithm [Cannon 1969] under the MPI specification [Özdogan 2006], because of its good

efficiency and low storage requirement. Using a similar method, we also implement Matrix-Vector Multiplication [Özdogan 2006] and Matrix Inversion [Pease 1967] using MPI. The second time-consuming part is the main loop for selecting instances greedily. To parallelize it, we employ the master-slave model [Huang and Wang 1997]. The master has two tasks: first, when loops begin, the master divides the whole work into independent jobs; second, master collects the result of slaves at the end of the loops and broadcasts updated information to all the machines. Specifically, master divides $U - S$ into disjoint subsets $S_1, \cdots, S_p$, and the i-th machine will start computing $C[v]$ in S_i and find the maxima concurrently; master waits for each slave to complete the current round and collects all the results to update for the next round.

5. EXPERIMENTS

In this section, we will evaluate the proposed approach for BMAL on both synthetic and real data sets. First, we use synthetic data sets to intuitively study our approach and to demonstrate the importance of combining all the three criteria. Then we validate our method on a real-world document classification data set. Finally, we study the efficiency performance of our parallel algorithm. All data sets, codes, and tools to analyze the results are publicly available. [1]

5.1. Synthetic Data Sets

The Gaussian Synthetic Data Set We use a synthetic data set to validate the effectiveness of our algorithm on the content information and to verify the necessity of all the three criteria. The Gaussian Synthetic Data Set is a data set with only content information. It is randomly generalized on a 2-D plane consisting of 17 gaussian distributions. Each distribution has been given to class label 0 or 1 randomly. The variance and number of points of each distribution are randomized as well. This synthetic data set has no link information.

We use this data set to demonstrate the necessity of all the three criteria. We design four tests using different definitions for the objective function representing different subsets of the criteria.

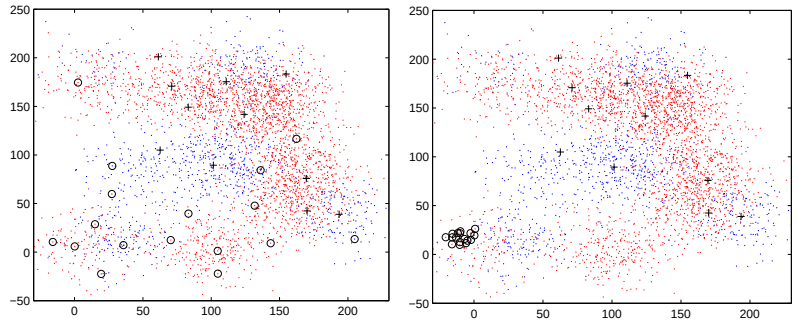
— Test1, **Proposed Method**: use our proposed data method, by equation 3.
— Test2, **Maximum Uncertainty**: apply the maximum uncertainty criterion. Let $\alpha = 0$, then the objective function is degraded to an entropy function, i.e., $Q = H(S)$.
— Test3, **Without Maximum Uncertainty**: do not consider the maximum entropy criterion. That is let $\alpha = 1$, and then the objective function is degraded to $Q = C(S)$.
— Test4, **Without Minimum Redundancy**: do not consider the minimum redundancy criterion. This test shows the subtlety in defining $C(S)$ to ensure the diversity. Use a modified definition of $C(S)$ as follows:

$$C(S) = \sum_{i \in U} s_i \sum_{j \in L \cup S} w_{ij}$$

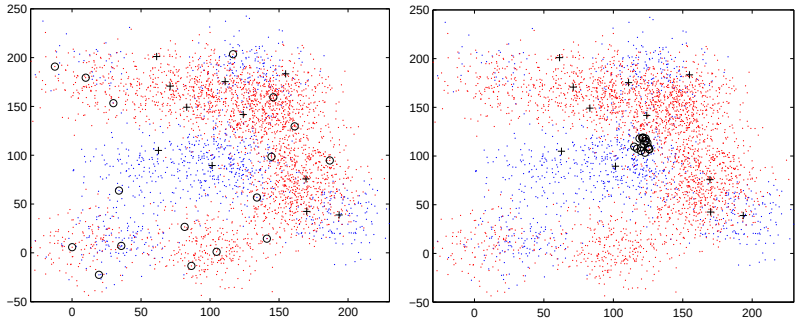
and keep the definition of $Q(S)$ to be the linear combination of $C(S)$ and $H(S)$. Note that this definition only substitutes the maximum operation with a summation operation thus still satisfies the maximum-uncertainty and maximum-impact criteria, except that the diversity property in Theorem 3.1 does not hold any more.

We use the simple KNN classifier ($k = 3$) to learn the classification model. Figure 3 shows the result of the tests. There are 3,612 data instances (points) in this synthetic

[1] http://arnetminer.org/cal/



(a) Test1(80.25%) (b) Test2(76.93%)



(c) Test3(75.45%) (d) Test4(69.36%)

Fig. 3. Four Tests on Gaussian Synthetic Data Set(accuracies are shown in bracket)

data on a 2-D plane and 11 of them are initially labeled. The task is to actively find another 17 data points for labeling. In Figure 3, red and blue points refer to two classes respectively. Initially labeled data points are displayed with "+", and the selected data points are labeled with "◯". From Figure 3, we can easily see that the selected data points in Test2 (3(b)) and Test4 (3(d)) are undesirable and their performance might be inferior than that in Test1 and Test3. This is because in Test2 and Test4, the selected instances are similar with each other (instances located together). The accuracy performance also confirms this observation. The four tests demonstrate that all the three criteria are necessary.

The Networked Synthetic Data Set In the networked synthetic data set, we only generate links without any content information. Specifically, in this data set, 385 data points in the graph are divided into 19 clusters. Each cluster is a star graph with a center surrounded by a random number of points, and all of the points in the cluster are in the same class. All of the data points have the same feature vector; in another word, only link information can be exploited in this graph. In figure 4, the initially labeled set is shown using "+", the selected data set using the proposed method is shown using "◯", and different classes are shown using different colors. We can see that it reasonably selects all the centers: it satisfies our intuition that the centers have the highest impact and they are distributed diversely enough.

In real data set, there are often noises and outliers. To further verify our accuracy under such scenarios, we bring two kinds of noises to our data set: two leaves in different clusters are connected with probability p_1; for each point in each cluster, it has probability p_2 to be labeled the opposite class. The error rate under different noise settings are shown in table II, where different rows stand for different p_2's, and differ-

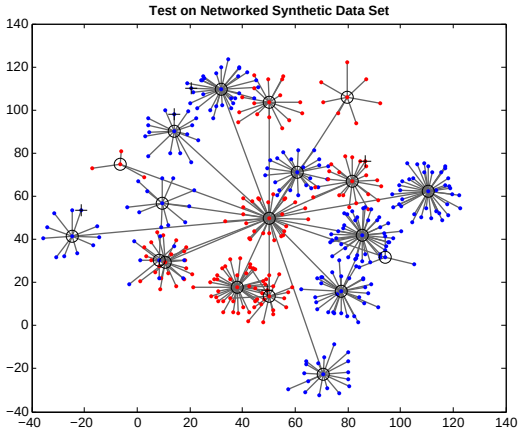


Fig. 4. Selection result on the Networked Synthetic Data Set Without Noises

ent columns stand for different p_1's. We can conclude that when noises are small, our model has a relatively small prediction error rate($< 10\%$); when noises get closer to 50% (which means the label of the nodes have nothing to do with the links), the error rate comes nearer to 50%.

Table II. Error rate(%) of Networked Synthetic Data

$p_2 \backslash p_1$	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
0.0	00.00	00.00	00.00	00.00	00.00	00.00
0.1	09.92	09.50	10.00	10.28	11.14	09.83
0.2	19.03	18.59	19.94	20.47	19.89	18.84
0.3	30.22	30.55	28.50	29.72	30.30	28.48
0.4	41.08	39.89	39.31	41.19	39.64	40.72
0.5	52.16	49.28	50.00	50.17	50.94	50.97

5.2. Real-World Data Sets With Links

Data Sets We use two citation data sets and one web page data set as the real-world data sets to test the proposed method for text classification. Statistics of the three data sets are given in Table III.

Cora Data Set [Yang et al. 2009; Sen et al. 2008] contains 2,708 scientific publications, and the documents are classified into seven categories (fields): i.e., Case Based, Genetic Algorithms, Neural Networks, Probabilistic Methods, Reinforcement Learning, Rule Learning, and Theory. After stemming and removing stop words and words that appear less than 10 times in the data set, we obtain a vocabulary of 1,433 unique words. There are 5,429 citation relationships between the documents.

Citeseer Data Set [Yang et al. 2009; Sen et al. 2008] contains 3,312 publications, categorized into 6 classes: Agents, AI, DB, IR, ML, and HCI. There are 3,703 unique words after processing, and the number of citations is 4,732.

WebKB Data Set [Sen et al. 2008] contains web pages from four computer science departments, and there are five categories: course, faculty, student, project, and staff. This is a subset of the original WebKB data set [Craven et al. 1998]. The webKB data set contains 877 web pages and 1,703 unique words. There are 2,868 total links between these pages.

For all the three data sets, we cast the multiple classification problem as multiple binary-classification tasks: for each category, we take documents of this category as positive instances and all the other documents as negative.

Table III. Real-World Data Sets with Links

Name	Topics	Documents	Citings	Words
Cora	7	2708	5429	1433
Citeseer	6	3312	4732	3703
WebKB	5	877	2868	1703

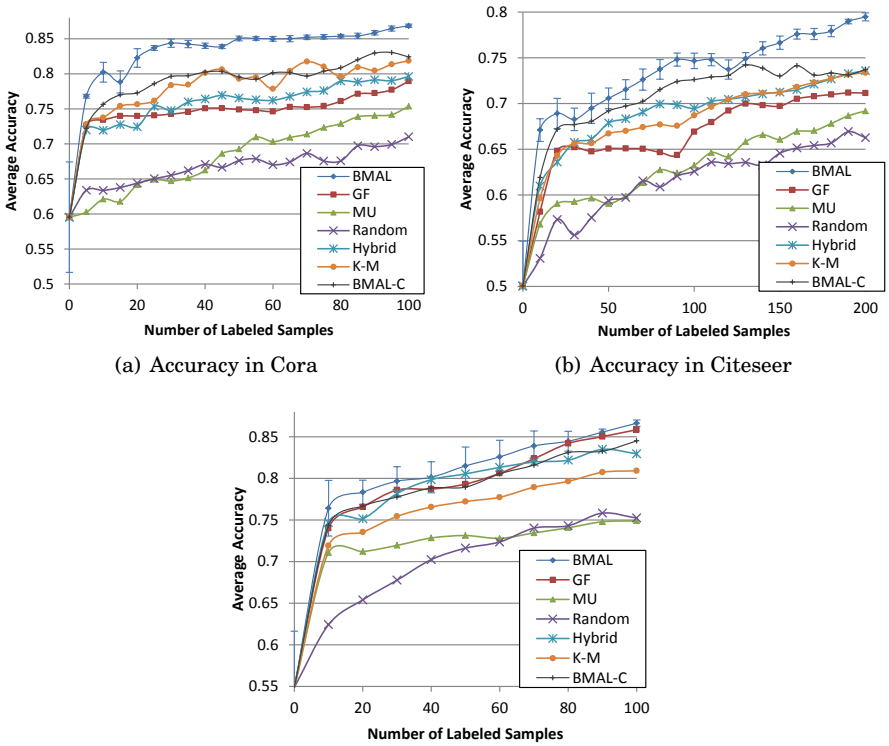
Baselines We define the following baseline methods:

— *Random*: selects the instances randomly with equal probability.
— *Most uncertainty*: selects the set with the largest entropy $H(S)$.
— *Active Learning using Gaussian Fields*: is an approach suggested by [Zhu et al. 2003] based on a semi-supervised learning framework using Gaussian fields and harmonic functions[Zhu et al. 2003]. As it is a single-mode active learning algorithm, we run this algorithm k times to select k instances. In this framework, the link information can be introduced using our proposed method in a similar way as in Section 3.2. We will utilize the link information in the tests.
— *Hybrid*: is suggested by [Macskassy 2009]. It asks for uncertainty approach and two graphical metrics(betweenness and cluster-finding) to find a selected set, and uses empirical risk to pick the best set among the union of the data instances selected by the three strategies.
— k-*means*: is suggested by [Rattigan et al. 2007]. In the article some active inference methods are compared with each other and k-means is found to be the best one among them. Here we employ the same strategy for active learning, that is, we find vertices using k-means as the labeled set and then train the classifier.

For simplicity, we use **Random, MU, GF, Hybrid, K-M** to denote the above baseline methods respectively. We refer to our model as **BMAL**.

Results We set the α parameter to be 0.5. For the Cora data set, we randomly pick 5 instances as the initially labeled set $\mathcal{L}$; for the other two data sets, this number is 10, due to the size of the data set and the learning difficulty. For the same reason, we define different batch size $k = 5, 10, 5$ for Cora, Citeseer and WebKB data sets respectively. The batch mode active learning methods first select k instances based on $\mathcal{L}$, and then repeatedly select k instances based on the union of initially labeled and selected instances; the single mode active learning methods iteratively select k instances, and repeatedly query the user and update the model. After the selecting process, we learn the classification model based on the selected data instances by different active learning methods using the same semi-supervised learning method. Here we use the NetKit-SRL toolkit [Macskassy and Provost 2007] to learn in networked data set. For each data set, we run the experiment 30 times with different initially labeled sets, and both the average and variance of the accuracy is used for final evaluation.

Figure 5 shows the results on each of the data set. Due to space limitations, we only draw the variance of the proposed method in the figure. The results show that maximum entropy does not have a good performance over all the data sets. In all the three data sets, our method outperforms the strategies based on graph metrics, i.e., the hybrid method and the k-means method. The performance of the Gaussian random field based method in the three results is a bit unstable: it does not perform well in the Cora and Citeseer data sets, while in the WebKB data set, its accuracy is close to the proposed method. In the WebKB data set, the gap between random selection and



(a) Accuracy in Cora

(b) Accuracy in Citeseer

(c) Accuracy in WebKB

Fig. 5. Tests on Three Data Sets with Links

these methods are not as high as other data sets, probably because it is not so easy in this web-linked data set to perform batch mode active learning. Also, from the view of variance, our method has average variances of 0.005, 0.009, 0.01 on the Cora, Citesser and WebKB data set, which are much smaller than the other methods.

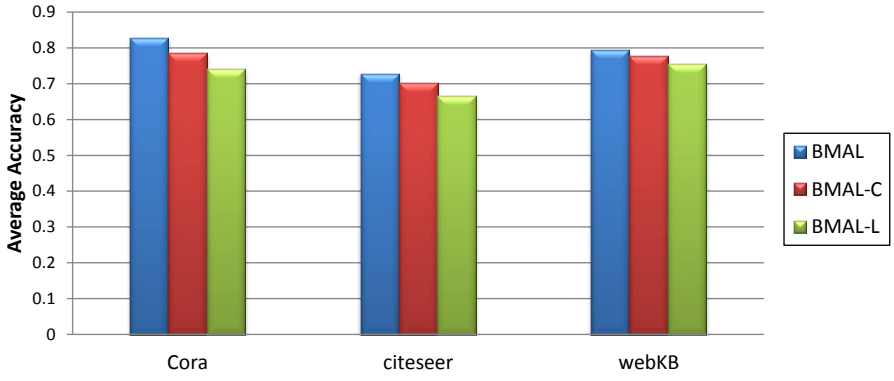
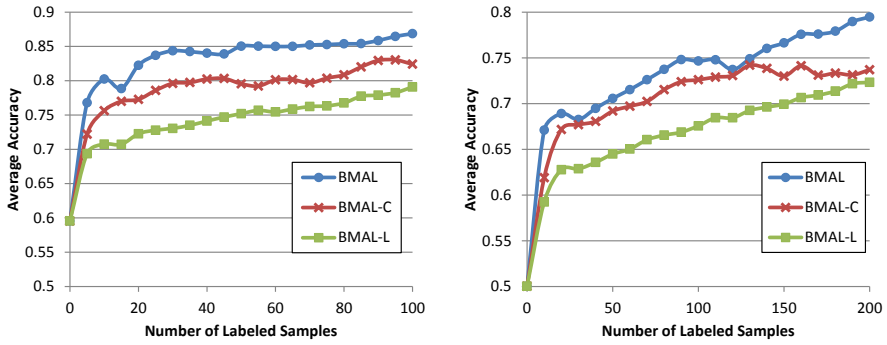
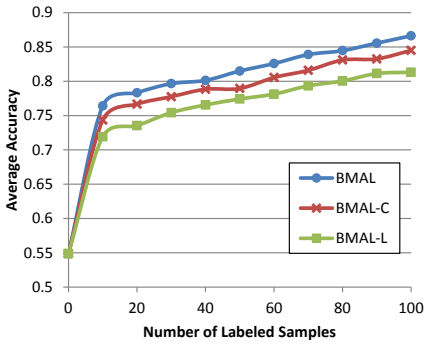


Fig. 6. Comparison of BMAL, BMAL-L, and BMAL-C



(a) Accuracy in Cora (b) Accuracy in Citeseer



(c) Accuracy in WebKB

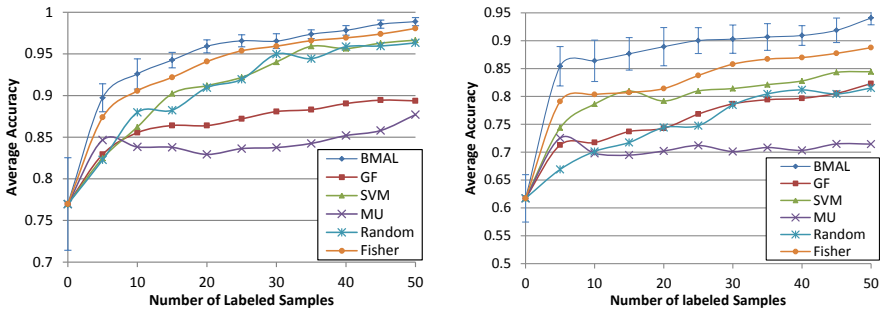
Fig. 7. Comparison of BMAL, BMAL-C, BMAL-L

Why BMAL outperforms others We empirically show that our BMAL method outperforms the baselines. The advantage of our algorithm is that it effectively combines the link and content information. We denote **BMAL-L** as a simplified version of our BMAL method without considering the content information and similarly **BMAL-C** as BMAL without link information. Figure 7 shows the accuracy of BMAL, BMAL-L and BMAL-C in different data sets and figure 6 shows the average accuracies of our method with different k's. Both have shown that combining link and content has greatly increased the classification result. Figure 5 also suggests that by removing the content information, the performance of our method decreases.

5.3. Real-World Data Set Without Links

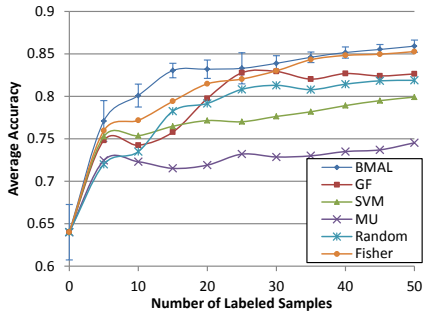
Experiment Setup We use the UCI 20 Newsgroup [Frank and Asuncion 2010] in this experiment. The data set contains approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different categories. Some of the categories are closely related to each other and some are unrelated. We construct four binary classification tasks:

— comp.sys.mac.hardware(963) vs comp.windows.x(988): 3338 words
— rec.sport.baseball(994) vs rec.sport.hockey(999): 3904 words
— comp.sys.ibm.pc.hardware(982) vs comp.sys.mac. hardware(963): 2812 words
— talk.religion.misc(628) vs alt.atheism(799): 3360 words

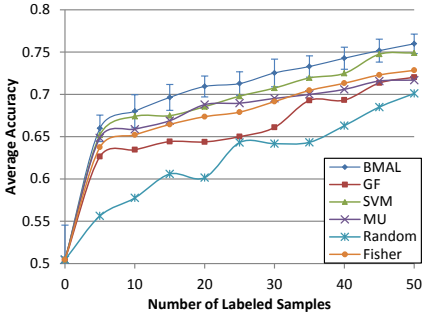


(a) Accuracy of Mac vs Windows

(b) Accuracy of Baseball vs Hockey



(c) Accuracy of IBM vs MAC



(d) Accuracy of Religion vs Atheism

Fig. 8. Tests on 20Newsgroups Data Set

The four classification tasks stand for different difficulty levels: the first one is easy, the last one is hard, and the left two stand for medium difficulty. In each task, we remove words which appear less than 10 times.

As for the baseline methods, besides Random, MU, GF, we introduce another two methods:

SVM: it is a batch mode active learning method, sampling examples that are closest to the decision boundary of SVM for the query [Tong and Chang 2001]. We use a modified version by [Hoi et al. 2008], which incorporates the diversity information as well. This method considers only content-based features.

Fisher: it is a batch mode active learning method based on information matrix. We choose the method proposed in [Hoi et al. 2006].

Other settings are similar to the experiments with links, except the learning method. We use the nearest subspace method as the classifier, for it is shown to have a good performance in document classification tasks[Li and Jain 1998].

Results Figure 8 shows the results of different methods. We only draw the variance of our proposed method because of space limitation,

From these results, we see that the performance of uncertainty is bad in the data set, even underperforms the random selection in three of the classification tasks. As demonstrated, maximum entropy tends to select similar instances, hence the accuracy is low. Also, the GF method does not perform as well as in the linked data sets. The SVM-based method is not stable: it has a bad performance in the task IBM vs MAC. Though the Fisher information based method has a high accuracy in the classification tasks, our method performs better than Fisher by accuracy. And our method has an

advantage in the variance in the experiments as well. In summary, we can see from the experiments that our method outperforms the baseline methods.

5.4. Efficiency Performance

Finally, we evaluate the efficiency performance of our parallel algorithm by comparing with the basic algorithm running on a single machine. First we run our parallel algorithm in a network with 2 computers, each with 4 cores. Table IV shows the running time comparison on the three data sets Citeseer, Cora and WebKB, which are already introduced in section 5.2. We also generate a large synthetic data set with 8,000 data instances to better demonstrate the time improvement. From the table we see that the parallel algorithm has a very good efficiency performance, for example, on the Citeseer data set it achieves a speedup of $\times 6$ and on the Synthetic data set $\times 7$.

Table IV. Time Comparison on Real-World Data Sets Running on 8 Cores

Data Set	Size	Basic CAL	Parallel CAL
WebKB	877	129.0s	36.6s
Cora	3312	481.1s	85.5s
Citeseer	2708	532.4s	88.7s
Synthetic	8000	3104.9s	437.2s

Figure 9 shows the speedup (the running time of basic algorithm divides the parallel algorithm) increases as the total number of cores increases. The dashed line is the ideal expectation of speedup. We can see from the figure that on large data sets, the speedup of the parallel algorithm is close to linear. On small data sets such as (selected) WebKB, the algorithm does not gain a good speedup performance, which is reasonable. Table IV lists the running time of our algorithm on different data sets.

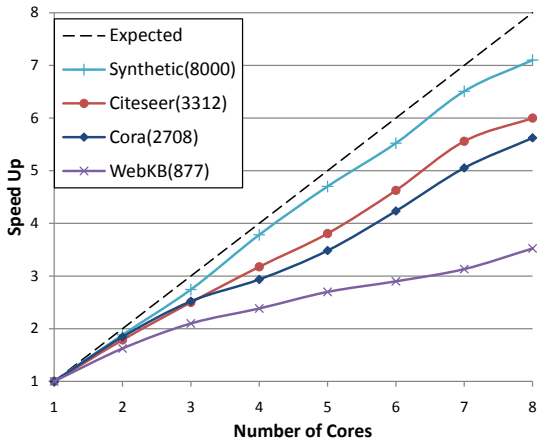


Fig. 9. Speedup v.s. Number of Cores

6. RELATED WORK

Batch Mode Active Learning : Active learning, or selective sampling, has been extensively studied for many years (See [Settles 2010] for a survey). However, classical single-mode active learning methods suffer from a couple of problems: First retraining is needed after each selection; Second most informative instances selected in each iteration might be highly correlated since redundancy cannot be taken into account. To solve this, a number of approaches have been proposed to perform active learning in batch mode. Generally SVM-based active learning methods [Tong and Chang 2001] repeatedly select the instance closest to the decision boundary, and redundancy is avoided by incorporating diversity [Brinker 2003]. Based on these researches, [Hoi et al. 2008] proposed a novel Min-Max SVM batch mode active learning framework. Another method utilizes the Fisher Information matrix. By setting Fisher Information Matrix as the objective function, a set of instances can be efficiently selected. This method has been applied to large scale text categorization[Hoi et al. 2006], medical image classification[Hoi et al. 2006] and image retrieval[Steven et al. 2009]. Rather than using heuristic measures, [Guo and Schuurmans 2008] directly learn a good classifier by formulating batch mode active learning as an optimization problem, and [Shi and Zhao 2010] tried to define active learning from classifier models. [Xu et al. 2009]. Compared with previous works, our work apply for networked data, aiming to give a clear set of criteria to optimize the result of active learning.

Classification in Networked Data : Recent focus in classification research extends to classify related entities by exploiting dependencies between data instances, which is shown to improve the accuracies under interrelated conditions [Jensen et al. 2004; Jensen and Neville 2002]. *Collective learning* is the fundamental approach when classifying such networked data sets. It was originally initiated by the work proposing iterative classification in relational data [Neville and Jensen 2000]. Earliest efforts towards learning in networked data sets include relaxation labeling [Hummel and Zucker 1983; Chakrabarti et al. 1998], iterated conditional modes[Besag 1986], etc. Since then many methods have been developed for collective learning, such as inductive logic programming[Slattery and Craven 1998], graph-cuts based formulation [Boykov et al. 2001], belief propagation in Probabilistic Relational Model[Taskar et al. 2001; Getoor et al. 2001; Taskar et al. 2002], iterative classification[Lu and Getoor 2003; Heß and Kushmerick 2004], region graph method[Yedidia et al. 2005] and so on. Collective Learning has applied to various topics such as text categorization [Namata et al. 2009], computer vision [Anguelov et al. 2005], social network [Liben-Nowell and Kleinberg 2007], etc.

Active Learning for Networked Data : Due to the success of collective inference on networked data, there have been proposals to extend it into Semi-Supervised learning scenario[Macskassy 2007; Xu et al. 2006]. Following this thread, several attempts have been done to apply active learning in networked data. Some works employ empirical risk minimization, which is shown to be computational expensive: [Roy and McCallum 2001] directly optimizes expected future error; [Zhu et al. 2003] combines semi-supervised learning and active learning based on Gaussian random fields and harmonic functions; [Macskassy 2009] integrates graph-metrics and empirical risk minimization. [Bilgic et al. 2010] proposes an active learning method for networked data built upon uncertainty sampling, committee-based sampling and clustering. Also, there have been explorations of active learning on special graphs, such as trees [Cesa-Bianchi et al. 2010]. In this paper we introduce a novel batch mode active learning framework which is independent of collective learning model and has been computationally efficient.

Active Inference : Active learning and active inference are similar in several ways. They both request classification label of a selected set from users, and improves the accuracy by designing how to select such samples. The fundamental difference is when these labels are collected: active learning request these labels at *training time*, while active inference request these labels at *prediction time* [Attenberg and Provost 2010]. As a result, when active learning framework requests labeling, the learning model is still training and updating; but when active inference framework will have a trained underlying classifier at the request time. Compared with active learning, active inference will result in lower cost, but the benefit of active labeling is limited in the sense that it cannot take effect to the trained classifier. Several approaches have been proposed about active inference in networked data: [Rattigan et al. 2007] introduced the active inference concept and study four different non-random selection methods based on network structure exploitation; [Bilgic and Getoor 2008] introduced a method based on objective function optimization; [Bilgic and Getoor 2009, 2010] proposed a novel framework of *reflect and correct*, by trying to find mistakes of underlying classifiers and correct them. Also, there has been active inference work in other kinds of data set, e.g. stream data [Attenberg and Provost 2010].

7. CONCLUSION

In this paper, we study a novel problem of batch mode active learning and propose a unified framework to solve this problem by combining both link and content information. We define three criteria to measure the informativeness of a set of data instances and design an objective function based on the three criteria. We demonstrate the effectiveness of the three criteria on synthetic data sets, and validate our proposed approach on several real-world data sets. Experimental results show that our approach outperforms several baseline methods for batch mode active learning. We have also developed a parallel implementation and validate its speedup performance on four data sets.

The general problem of batch mode active learning represents a new and interesting research direction in machine learning and data mining. There are many potential future directions of this work. A direct adaptation is to apply the proposed method to active learning for link prediction, an important learning problem over networks. Another interesting issue is to study how to combine the active learning process and the classification model learning process together. Currently, the active learning is considered as an independent process from the classification process. However, they are usually intertwined. Combining the two processes together may further improve the classification accuracy. Another potential research is to apply the proposed approach to other applications on social networks (e.g., social influence analysis [Tang et al. 2009]), a very important application scenario on the Web to further validate its effectiveness.

REFERENCES

ANGUELOV, D., TASKAR, B., CHATALBASHEV, V., KOLLER, D., GUPTA, D., HEITZ, G., AND NG, A. 2005. Discriminative learning of markov random fields for segmentation of 3d scan data. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 169–176.

ATTENBERG, J. AND PROVOST, F. 2010. Active inference and learning for classifying streams. In *Proceedings of Budgeted Learning Workshop in International Conference on Machine Learning(ICML Workshop)*.

BESAG, J. 1986. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 259 – 302.

BEYGELZIMER, A., DASGUPTA, S., AND LANGFORD, J. 2009. Importance weighted

active learning. In *Proceedings of the International Conference on Machine Learning(ICML)*. 49–56.

BILGIC, M. AND GETOOR, L. 2008. Effective label acquisition for collective classification. In *ACM Special Interest Group on Knowledge Discovery and Data Mining(SIGKDD)*. 43–51.

BILGIC, M. AND GETOOR, L. 2009. Reflect and correct: A misclassification prediction approach to active inference. *ACM Transactions on Knowledge Discovery from Data(TKDD) 3,* 4, 1–32.

BILGIC, M. AND GETOOR, L. 2010. Active inference for collective classification. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*.

BILGIC, M., MIHALKOVA, L., AND GETOOR, L. 2010. Active learning for networked data. In *Proceedings of International Conference on Machine Learning (ICML)*.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI) 23*, 1222–1239.

BRINKER, K. 2003. Incorporating diversity in active learning with support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*. 59–66.

CANNON, L. E. 1969. A cellular computer to implement the kalman filter algorithm. Ph.D. thesis, Montana State University.

CESA-BIANCHI, N., GENTILE, C., VITALE, F., AND ZAPPELLA, G. 2010. Active learning on trees and graphs. Tech. rep., MIT Press.

CHAKRABARTI, S., DOM, B., AND INDYK, P. 1998. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the ACM Special Interest Group on on Management of data(SIGMOD)*. 307–318.

CRAVEN, M., DIPASQUO, D., FREITAG, D., MCCALLUM, A., MITCHELL, T., NIGAM, K., AND SLATTERY, S. 1998. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*. 509–516.

FRANK, A. AND ASUNCION, A. 2010. UCI machine learning repository.

GETOOR, L., SEGAL, E., TASKAR, B., AND KOLLER, D. 2001. Probabilistic models of text and link structure for hypertext classification. In *International Joint Conference on Artificial Intelligence Workshop on "Text Learning: Beyond Supervision" (IJCAI Workshop)*. 24 – 29.

GROPP, W., LUSK, E., AND SKJELLUM, A. 1994. *Using MPI: Portable Parallel Programming with the Message Passing Interface*. the MIT Press.

GUO, Y. AND SCHUURMANS, D. 2008. Discriminative batch mode active learning. In *Proceedings of Advances in Neural Information Processing Systems(NIPS)*. 593–600.

HARPALE, A. S. AND YANG, Y. 2008. Personalized active learning for collaborative filtering. In *Proceedings of Special Interest Group on Information Retrieval(SIGIR)*. 91–98.

HESS, A. AND KUSHMERICK, N. 2004. Iterative ensemble classification for relational data: A case study of semantic web services. In *Proceedings of European Conference on Machine Learning(ECML)*. 156–167.

HOI, S. C. H., JIN, R., AND LYU, M. R. 2006. Large-scale text categorization by batch mode active learning. In *Proceedings of the World Wide Web Conference(WWW)*. 633–642.

HOI, S. C. H., JIN, R., ZHU, J., AND LYU, M. R. 2006. Batch mode active learning and its application to medical image classification. In *Proceedings of the International Conference on Machine Learning(ICML)*. 417–424.

HOI, S. C. H., JIN, R., ZHU, J., AND LYU, M. R. 2008. Semi-supervised svm batch mode active learning for image retrieval. In *Proceedings of IEEE Conference on*

Computer Vision and Pattern Recognition(CVPR). 1–7.
HUANG, K. AND WANG, F. 1997. Design patterns for parallel computations of master-slave model. In *Proceedings of International Conference on Information, Communications and Signal Processing (ICICS)*. 1508 – 1512.
HUMMEL, R. AND ZUCKER, S. 1983. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI) 5,* 3, 267 – 287.
JENSEN, D. AND NEVILLE, J. 2002. Linkage and autocorrelation cause feature selection bias in relational learning. In *Proceedings of the International Conference on Machine Learning(ICML)*. 259–266.
JENSEN, D., NEVILLE, J., AND GALLAGHER, B. 2004. Why collective inference improves relational classification. In *Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*. 593–598.
JOSHI, A. J., PORIKLI, F., AND PAPANIKOLOPOULOS, N. 2010. Multi-class batch-mode active learning for image classification. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 1873–1878.
KAWAHARA, Y., NAGANO, K., TSUDA, K., AND BILMES, J. 2009. Submodularity cuts and applications. In *Proceedings of Advances in Neural Information Processing Systems(NIPS)*.
KSCHISCHANG, F. R., FREY, B. J., AND LOELIGER, H. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory 47*, 498–519.
LAFFERTY, J. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning(ICML)*. 282–289.
LI, Y. AND JAIN, A. K. 1998. Classification of text documents. In *Proceedings of International Conference on Pattern Recognition (ICPR)*. 1295.
LIBEN-NOWELL, D. AND KLEINBERG, J. 2007. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology (JASIST) 58,* 7, 1019–1031.
LU, Q. AND GETOOR, L. 2003. Link-based classification. In *Proceedings of the 20th International Conference on Machine Learning(ICML)*. 496 – 503.
MACSKASSY, S. A. 2007. Improving learning in networked data by combining explicit and mined links. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*. 590–595.
MACSKASSY, S. A. 2009. Using graph-based metrics with empirical risk minimization to speed up active learning on networked data. In *Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*. 597–606.
MACSKASSY, S. A. AND PROVOST, F. 2007. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research (JMLR) 8*, 935–983.
NAMATA, G. M., SEN, P., BILGIC, M., AND GETOOR, L. 2009. Collective classification for text classification. In *Text Mining: Classification, Clustering, and Applications*. Taylor and Francis Group.
NEMHAUSER, G., WOLSEY, L., AND FISHER, M. 1978. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming 14*, 265–294.
NEVILLE, J. AND JENSEN, D. 2000. Iterative classificataion in relational data. In *Proceeding of National Conference on Artificial Intelligence Workshop on Statistical Relational Learing (AAAI Workshop)*. 42 – 49.
NODELMAN, U., SHELTON, C., AND KOLLER, D. 2003. Learning continuous time bayesian networks. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence(UAI)*. 451–458.

ÖZDOGAN, C. 2006. Cannon's matrix-matrix multiplication with mpi's topologies. siber.cankaya.edu.tr/ozdogan/GraduateParallelComputing/ceng505/.

PEASE, M. C. 1967. Matrix inversion using parallel processing. *Journal of ACM 14,* 4, 757–764.

RAJAN, S., YANKOV, D., GAFFNEY, S. J., AND RATNAPARKHI, A. 2010. A large-scale active learning system for topical categorization on the web. In *Proceedings of the World Wide Web Conference(WWW)*. 791–800.

RATTIGAN, M. J., MAIER, M., AND JJENSEN, D. 2007. Exploiting network structure for active inference in collective classification. In *Proceedings of International Conference on Data Mining Workshop (ICDM Workshop)*. 429–434.

ROY, N. AND MCCALLUM, A. 2001. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*. 441–448.

SEN, P., NAMATA, G. M., BILGIC, M., GETOOR, L., GALLAGHER, B., AND ELIASSI-RAD, T. 2008. Collective classification in network data. *AI Magazine 29,* 3, 93–106.

SETTLES, B. 2010. Active learning literture survey. Tech. Rep. 1648, Computer Science Department, University of Wisconsin-Madison.

SHI, L. AND ZHAO, Y. 2010. Batch mode sparse active learning. In *Proceedings of International Conference on Data Mining Workshop (ICDM Workshop)*. 875–882.

SLATTERY, S. AND CRAVEN, M. 1998. Combining statistical and relational methods for learning in hypertext domains. In *In Proceedings of the 8th international Conference on Inductive Logic Programming (ILP)*. 38–52.

STEVEN, C. H. H., RONG, J., AND R, R. L. M. 2009. Batch mode active learning with applications to text categorization and image retrieval. *IEEE Transactions on Knowledge and Data Engineering (TKDE) 21*, 1233–1248.

TAN, C., TANG, J., SUN, J., LIN, Q., AND WANG, F. 2010. Social action tracking via noise tolerant time-varying factor graphs. In *Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*. 1049–1058.

TANG, J., SUN, J., WANG, C., AND YANG, Z. 2009. Social influence analysis in large-scale networks. In *Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*. 807–816.

TASKAR, B., ABBEEL, P., AND D.KOOLER. 2002. Discriminative probabilistic models for relational data. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence(UAI)*. 485 – 492.

TASKAR, B., SEGAL, E., AND KOLLER, D. 2001. Probabilistic classification and clustering in relational data. In *Proceedings of International Joint Conference on Artificial Intelligence(IJCAI)*. 870–878.

TONG, S. AND CHANG, E. 2001. Support vector machine active learning for image retrieval. In *Proceedings of ACM Multimedia(MULTIMEDIA)*. 107–118.

XU, L., WILKINSON, D., SOUTHEY, F., AND SCHUURMANS, D. 2006. Discriminative unsupervised learning of structured predictors. In *Proceedings of the International Conference on Machine Learning(ICML)*. 1057–1064.

XU, Z., HOGAN, C., AND BAUER, R. 2009. Greedy is not enough: An efficient batch mode active learning algorithm. In *Proceedings of International Conference on Data Mining Workshop (ICDM Worshop)*. 326–331.

YANG, T., JIN, R., CHI, Y., AND ZHU, S. 2009. Combining link and content for community detection: a discriminative approach. In *Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*. 927–936.

YEDIDIA, J., FREEMAN, W., AND WEISS, Y. 2005. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory 51,* 7, 2282–2312.

ZHU, X. 2005. Semi-supervised learning with graphs. Ph.D. thesis, Carnegie Mellon University. CMU-LTI-05-192.

ZHU, X., GHAHRAMANI, Z., AND LAFFERTY, J. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the International Conference on Machine Learning(ICML)*. 912–919.

ZHU, X., LAFFERTY, J., AND GHAHRAMANI, Z. 2003. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the International Conference on Machine Learning Workshop (ICML Workshop)*. 58–65.

A. APPENDIX: THE PROOF OF REDUNDANCY

We will prove a result for general definition of similarity matrix not limited to RBF: For a similarity matrix which is (λ_1, λ_2)-transitive(defined in section 3.2), and with the same definition of $R(S)$ as equation 9 , we have that: if $\forall j \in S, \exists i \in U - S, j = dp(i)$,

$$(1)\ R(S) \geq \frac{k+l}{n-k} \sum_{i \in U-S} \left(\frac{1}{w_{i,dp(i)}} \sum_{j \in S \cup L} w_{ij}^{\frac{1}{\lambda_2}} \right) - (k+l)$$

$$(2)\ R(S) \leq \frac{k+l}{n-k} \sum_{i \in U-S} \left(\frac{1}{w_{i,dp(i)}} \sum_{j \in S \cup L} w_{ij}^{\frac{1}{\lambda_1}} \right) - (k+l)$$

PROOF. We will only prove the first inequity, the proof of the second one is very similar.

For $\forall i \in U - S$, $\forall j \in S \cup L - \{dp(i)\}$, where $dp(i)$ is defined in equation 8. By the approximately transitive property, we have $w_{ik} \leq (w_{i,dp(i)} w_{dp(i),j})^{\lambda_2}$, hence

$$w_{dp(i),j} \geq w_{ij}^{\frac{1}{\lambda_2}} \frac{1}{w_{i,dp(i)}} \tag{11}$$

Sum equation 11 over all j's in $S \cup L - \{dp(i)\}$, we have

$$\sum_{j \in S \cup L - \{dp(i)\}} w_{dp(i),j} \geq \frac{1}{w_{i,dp(i)}} \sum_{j \in S \cup L - \{dp(i)\}} w_{ij}^{\frac{1}{\lambda_2}} \tag{12}$$

Since $\lambda_2 \leq 1$,

$$\sum_{j \in S \cup L - \{dp(i)\}} w_{dp(i),j} \geq \left(\frac{1}{w_{i,dp(i)}} \sum_{j \in S \cup L} w_{ij}^{\frac{1}{\lambda_2}} \right) - 1 \tag{13}$$

Now for each $j \in S$, we can see that $dp^{-1}(j) \neq \emptyset$. Let

$$I = \left\{ i : i = \operatorname{argmax} \frac{\sum_{j \in S \cup L} w_{ij}^{\frac{1}{\lambda_2}}}{w_{i,j_0}}, j_0 \in dp^{-1}(i) \right\} \tag{14}$$

Sum Eq. 13 over all $i \in I$,

$$\sum_{i,j\in S, i\neq j} w_{ij} = \sum_{i\in I}\left(\frac{1}{w_{i,dp(i)}}\sum_{j\in S\cup L} w_{ij}^{\frac{1}{\lambda_2}}\right) - (k+l) \tag{15}$$

By the definition of I by equation 14, we can see that the average of $\frac{\sum_{j\in S\cup L} w_{ij}^{\frac{1}{\lambda_2}}}{w_{i,dp(i)}}$ over I should be higher than the average over all $i \in U - S$'s, or,

$$\frac{\sum_{i\in I}\left(\frac{1}{w_{i,dp(i)}}\sum_{j\in S\cup L} w_{ij}^{\frac{1}{\lambda_2}}\right)}{k+l} \geq \frac{\sum_{i\in U-S}\left(\frac{1}{w_{i,dp(i)}}\sum_{j\in S\cup L} w_{ij}^{\frac{1}{\lambda_2}}\right)}{n-k} \tag{16}$$

Therefore $R(S) \geq \frac{k+l}{n-k}\sum_{i\in U-S}\left(\frac{1}{w_{i,dp(i)}}\sum_{j\in S\cup L} w_{ij}^{\frac{1}{\lambda}_2}\right) - (k+l)$ □

We now present lemma A.1 to prove theorem 3.1:

LEMMA A.1. *The RBF definition of W (equation 1) is $(2,1)$-transitive.*

PROOF. The following inequity holds for quadratic optimization:

$$\frac{1}{2}||\boldsymbol{x}_i - \boldsymbol{x}_k||^2 \leq ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 + ||\boldsymbol{x}_j - \boldsymbol{x}_k||^2 \leq ||\boldsymbol{x}_i - \boldsymbol{x}_k||^2 \tag{17}$$

The lemma can be directly proven from equation 17. □